

MBED を用いたカードリーダーシステムの開発

総合技術センター

計測・制御技術分野 七條 香緒莉 (Kaori Shichijo)

1. はじめに

本学において、学生証を使用した非接触カードリーダーによる入退室管理システムを採用している講義室や研究室が見受けられる。この度、同等のシステム開発に携わる機会を与えられた。本稿ではその内容について報告する。

2. MBED について

システムの根幹となるマイコンモジュールについては、MBED を採用した。今までに研修等で触れる機会があったことや、各種通信機構など様々な機能がデフォルトで搭載されていることが主な理由である。

特徴として、開発環境は図 1 に示すような web ブラウザによるクラウド環境となる。公式サイトからアカウント登録し、無料でワークスペースを設けることができる。そのため、インターネット接続環境があるなら時間や場所を問わず開発が可能となるのが大きな利点である。サイトは SNS のようになっており、Q&A のやり取りや自作のプログラムを公開する等、ユーザー同士での交流も可能である。

反面、サーバに障害が発生したときにはビルドが実行できなかったり、ソースファイルの閲覧すら不可能になったりすること

もある。実際、今回の開発中にも何度かそのような状況に陥った。大抵の場合は時間を空けることで解決するが、ソフトウェアをインストールして環境を整えればオフラインでの作業も可能となる。なお、本開発においてはオフライン環境を使用することはなかった。

3. IC カードとの通信

非接触 IC カードにはコイル状のアンテナが内蔵されている。それをカードリーダーモジュール（以下 CR）が発する電磁波にかざすことで電磁誘導を発生させ、内部の IC チップに電流を供給し CR との通信を行う。カードの規格には複数の種類があるが、今回は学生証を用いるということで必然的に MIFARE を採用することになった。

CR と MBED との通信はシリアル通信（UART）で行う。フレームと呼ばれる 16 進数のデータのまとまりを送受信し、カードに登録された学生番号等の情報を読み出していく。当然それらの情報は重要な個人情報となるので、鍵を用いた認証を繰り返す必要がある。

図 2 に本システムの簡単な流れを示す。図中のカード ID とは、各カードに与えられている固有の番号である。

4. プログラム制作

ここでは、開発において特に印象に残った点、苦勞した点を述べる。

4. 1 共有ライブラリ

前述のとおり、MBED の公式サイトには様々なプログラムが掲載され、初心者でもそれを利用して気軽に MBED を用いたプログラム開発ができる。公式から提供されているものに加え、世界各国における数多くのユーザーが自作のソースコードを無料で公開している。ごく限られた動作を行うプ

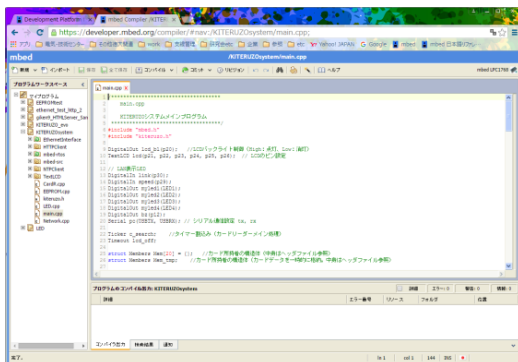


図 1 開発環境

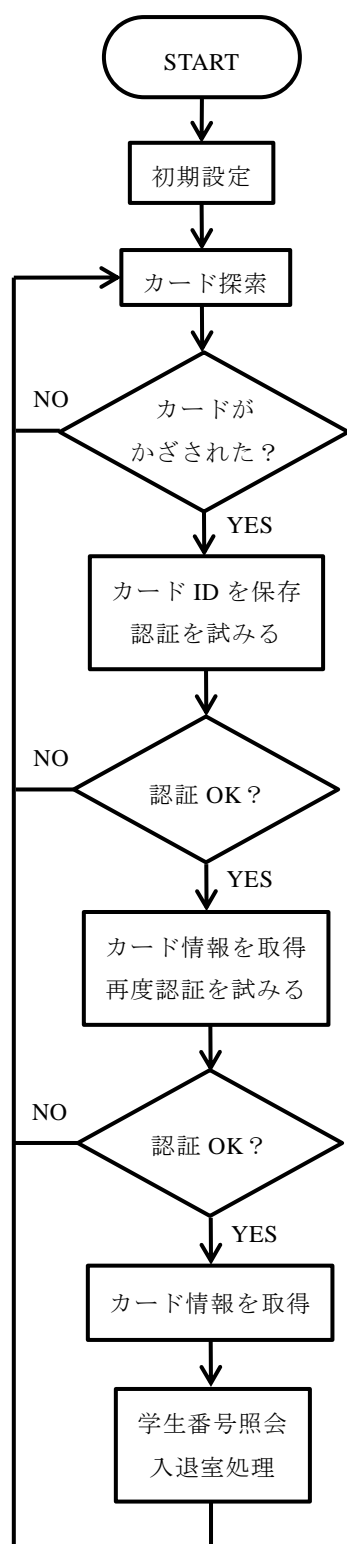


図2 システムの流れ

プログラム単体もあれば、ネットワーク通信やシリアル通信等、汎用的な動作に必要なプログラムをまとめたライブラリと呼ばれるものもある。それらは自身のワークスペースに自由にインポートして使用することが可能である。ライブラリを予めプログラ

ムに組み込めば、該当する通信や割込み等の複雑な動作の実現が、簡単なソースの記述だけで可能となる。

しかし、利点ばかりではない。ソースコードというものは、同じ動作を実現しようとしても書き手によって内容がそれぞれ全く異なるものになってしまうこともままある。そのため、各ソースコードの必要な箇所を部分的に参考にしたプログラムを制作しても、それぞれが上手く噛み合わず最終的に正しく動作しないといった問題が起こりうる。中には未完成のままライブラリとして公開されているものもあり、その場合は使用者が適宜修正・改良する必要がある。やはり最終的には自身の目的に合った独自の形に作り上げなければならない。

ただ、参考となるプログラムの有無で開発時間が大幅に変わるのは事実である。本開発においても公開されているソースコードは大変参考になった。

4. 2 非セレクト化とリトライ回数

CR にカードをかざすと通信が行われるのは前述のとおりである。情報を読み終えたカードとはそれ以降の通信が必要ないので、コマンドを送信し通信終了を通知する。これを非セレクト化という。非セレクト化しない場合、カードをかざし続ける限り連続でカードを検出してしまふ。非セレクト化することで一度認識したカードは検出なくなり、一旦カードを CR から遠ざけることで再度検出が可能となる。カードがかざされた時刻や回数は入退室を管理する上で大変重要な要素なので、非セレクト化しないことにはシステムが成り立たない。

本システムにおいても非セレクト化を行ったのだが、非セレクト化したにもかかわらず、その後 CR が間を空けずカードを検出してしまふという予期せぬ事態が起きた。開発においては MBED と CR 間の通信を記録する機器を接続していたため、通信内容は把握できていた。それによると、非セレクト化を指示するコマンドに対して、その成功を知らせるコマンドが確かに返されていたにもかかわらず、非セレクト化直後に再度同じカードを検出してしまふ。原因は

すぐに判明せず、非セレクト化後数秒間はカード探索を止めるという形で一時的に対応した。しかしこの場合、入退室者がそのままカードをかざし続ければ、探索を再開した瞬間に検出してしまう。逆に、停止している間に次のカードがかざされると、それにはすぐに対応できない。

そのような問題を抱えたままだが、一部にとらわれ続けるのもよくないので、取り敢えず別の部分の開発に移っていた。そこで、カード探索のコマンドを送信した際、カードをかざしていない場合の CR からの返信が、図 3 に示すようになっていることに気付いた。図中において、上の段に表示された 16 進数の並びが MBED から CR への送信コマンドであり、下の段がそれに対する CR の返信、つまり MBED の受信コマンドである。受信コマンドに着目すると、コマンドを無事受信したことを意味する ACK フレームと呼ばれるものだけである。本来はこの後に、検出したカードが 0 枚である旨を通知する Normal フレームが付随していなければならない。

改めて CR の仕様書を見直してみると、リトライ回数という言葉が目に残った。これはカード探索の際にカードを検出できなかったとき、再度探索を試みる回数を意味する。この値がデフォルトでは無限回数を意味する値となっていた。つまり、カードを検出するまで結果を返信せず探索し続けるという仕様だった。これをリトライしないように変更すると、図 4 に示すように、カードを検出できなかった場合に検出数が 0 である旨を通知する Normal フレームが付

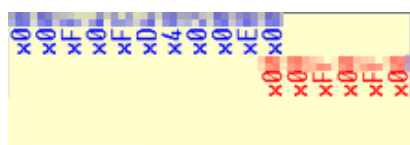


図 3 ACK フレームのみの返信

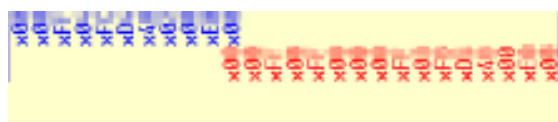


図 4 Normal フレームが付随した返信

随するようになった。

更に、この変更の結果、前述した非セレクト化の問題も解決した。リトライ回数を 0 としたことで非セレクト化が機能し、カードを一旦 CR から遠ざけない限り、初回以降はカードを検出しなくなったのだ。すなわち、非セレクト化しても無限に探索することで、そのうちカードを検出してしまっていたものと思われる。

4. 3 デバッグ作業

システム開発において、デバッグ作業は避けて通れない。よくある方法としては、PC のコンソールウィンドウ上に文字を出力するのが一般的である。ソースコードの各ポイントに文字出力関数を配置して動作の流れが正しいか確認したり、CR から受信したコマンドをそのまま表示させて正常に受信できているか確認したりと、本開発でも非常に役に立った。

ただし、文字を出力するという動作自体もプログラムの一部に含まれるということを忘れてはならない。ウィンドウへの出力があまりに増えすぎてしまうと、その動作で手一杯になり正規の動作に悪影響を及ぼしてしまう。本開発でも画面出力の量が多すぎたせいで処理に負荷がかかり、肝心のカード探索動作が正常に機能しなくなるトラブルに見舞われた。

5. まとめ

最終的に本システムは期待通りの動作をほぼ実現し、順調に稼働している。また、現在は別のモジュールを用いた同様のシステムを開発する機会に恵まれているが、今回の経験が大いに役立っていると実感している。

謝辞

共に本開発に携わり多大なるご協力を賜りました飯田仁技術職員、三浦隆浩技術職員両名に感謝申し上げます。